Qualitative Structural Analysis Using Diagrammatic Reasoning

Shirley Tessler
Computer Industry Project
Stanford University
Stanford, CA 94305

Yumi Iwasaki
Knowledge Systems Lab.
Stanford University
701 Welch Road
Palo Alto, CA 94304

Kincho Law
Dept. of Civil Engineering
Stanford University
Stanford, CA 94305

Abstract

Diagrammatic reasoning is a type of reasoning in which the primary means of inference is direct manipulation and inspection of a diagram. Diagrammatic reasoning is prevalent in human problem solving behavior, especially for problems involving spatial relations among physical objects. Our research examines the relationship between diagrammatic and symbolic reasoning in a computational framework. We have built a system, called REDRAW, that emulates the human capability for reasoning with pictures in civil engineering. The class of structural analysis problems chosen provides a realistic domain whose solution process requires domain-specific knowledge as well as pictorial reasoning skills. We hypothesize that diagrammatic representations provide an environment where inferences about the physical results of proposed structural configurations can take place in a more intuitive manner than that possible through purely symbolic representations.

1 Introduction

Humans often use diagrams to facilitate problem solving. In many types of problems, including but not limited to problems involving behaviors of physical objects, drawing a diagram is a crucial step in the solution process. A drawing can reveal important information that may not be explicit in a written description, and can help one gain insights into the nature of the problem. Though such use of diagrams is an integral part of human problem solving behavior, it has not received nearly as much attention in AI as symbolic reasoning has.

One important advantage of diagrammatic representation is that it makes explicit the spatial relations that might require extensive search and numerous inference steps to discover using a symbolic representation. Larkin and Simon have shown that, even when the information contents of symbolic and diagrammatic representations are equivalent, a diagrammatic representation can offer computational advantage in problems where spatial relations play a prominent role [Larkin and Simon, 1987], Since humans reason with so much apparent ease in some problems, a program that could reason directly with a diagrammatic representation would be more understandable to the user than a program that reasons exclusively with a purely symbolic representation. In addition, a diagrammatic reasoning program could offer insight into the relation between diagrammatic and symbolic reasoning, and may also be useful in imparting visualization skills to students of various engineering disciplines where such facility is crucial.

In this paper, we present our work aimed towards understanding the role of diagrammatic reasoning in problem solving. The problem we chose for studying diagrammatic reasoning is that of determining the deflection shape of a building frame structure under load. We have constructed a computer program called REDRAW (Reasoning with Drawings) that solves this problem qualitatively using a diagram in a way similar to human engineers.

1.1 Roles of Diagrams in Problem Solving

Some research has been done on the roles that diagrammatic reasoning play in human problem solving. Novak and Bulko [Novak and Bulko, 1992] have asserted that a diagram and its annotations serve as a short-term memory device in the problem solving process. Such a device allows temporarily-needed information to be retrieved later in the same manner that writing down intermediate results in multiplication problems frees the person to perform further calculations. They also postulate that a diagram may act as a substrate or concept anchor that allows the new part of a problem to be described relative to another well-understood problem. Larkin and Simon discuss extensively the advantages of diagrams for facilitating inference about topological or geometric relations [Larkin and Simon, 1987]. Chandrasekaran and Narayanan [Chandrasekaran and Narayanan, 1992], Novak and Bulko [Novak and Bulko, 1992], and Borning [Borning, 1979] have also pointed out the usefulness of diagrams to human problem solvers as a device to aid in visualization, "gedanken experiments" or prediction. Finally, Novak and Bulko [Novak and Bulko, 1992], and Koedinger [Koendinger, 1992] have explored the idea that diagrams may sometimes be used not primarily for making base-level inference, but rather to help in the selection of an appropriate method to solve a problem; that is, as an "aid in the organization of cognitive activity" [Chandrasekaran *et al.*, 1993],

A salient feature of diagrammatic reasoning in many situations is its qualitativeness. People reason with diagrams to get rough, qualitative answers. Though they must resort to formal, mathematical techniques for a more precise, quantitative answer, qualitative techniques are extremely useful in gaining valuable insight into the range of possible solutions. An initial qualitative understanding thus obtained can guide later analysis for more detailed answers. In the context of structural analysis, knowing the qualitative deflected shape allows one to identify critical features of the shape. One can then set up relevant equations in order to obtain more precise information such as actual magnitudes of forces and displacements at specific points of interest.

How do diagrams actually help civil engineers make qualitative inferences? From studying textbooks on elementary structural analysis, such as [Brohn, 1984], which aim to develop a intuitive understanding of the response of the structure under a load, we find that diagrams fulfill many of the same roles mentioned above. First, diagrams are used as "a visual language of structural behavior that can be understood with the minimum of textual comments" [Brohn, 1984]. The language allows engineers to express explicitly the constraints or physical laws that are relevant at each part of the proposed structure, in such a way that the constraints and some of the consequences are immediately apparent without further reasoning. Secondly, the diagram serves as a place holder or short-term memory device by allowing engineers to sketch out the result of one deformation and then go back to see if there is a further effect or interaction that needs to be addressed. Finally, visual inspection of diagrams guides engineers in choosing the next step, resulting in a more efficient problem solving process than it would be otherwise. REDRAW uses diagrams in all these capacities in the context of determining the deformation shape of frame structures.

2 Deflection Shape Problem

Determining the qualitative deflected shape of a frame structure under load is a crucial step in analyzing the behavior of a structure. Figure 1 shows the reasoning steps an engineer may go through using a diagram. She first makes a simple, 2-D drawing of the shape of the given frame structure (Figure 1(a)). Given a load on the structure, she modifies the shape of the structural member under the load (b). She inspects the modified shape to identify the places where constraints for equilibrium of the structure are violated. Those constraint violations are corrected by modifying the shape of connected structural members, propagating deflection to other parts of the structure (c). This process is repeated until all the constraints are satisfied (d - f). The drawing thus produced (f) shows the final deflected shape of the frame under the given load.

Given a diagram of a frame structure and a load, REDRAW produces an underlying symbolic representation in order to facilitate reasoning about engineering concepts. Then the program will use its structural engineering knowledge to propagate constraints on the diagram of the structure and will inspect and modify this picture until a final shape is produced that represents a stable deflected structure under the given load.

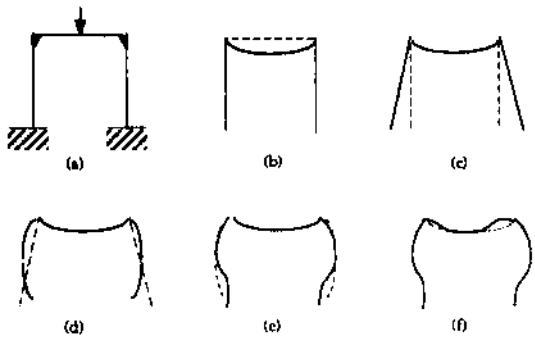


Figure 1: Steps in determining the deflected shape

As with the qualitative nature of human visual reasoning, the reasoning carried out by REDRAW is also qualitative. The answer it produces is a picture of a deflected shape. Although the resulting picture is qualitatively consonant with the problem solution, it is not (nor does it need to be) mathematically accurate. In fact, the magnitude of deflections in REDRAW's diagrams is much larger than actual magnitudes. This over-emphasis is done intentionally, since the deflections in actual structures would be too small to see in a simple diagram, and since people almost always exaggerate features of interest when they draw this type of diagrams.

REDRAW solves this type of deflected shape problem by directly manipulating a representation of the shape in a similar manner to that shown above. Although the problem could be solved by setting up equations, visualization is the indispensable first step that provides an engineer with intuitive understanding of the behavior of the structure and enables her to recognize a good strategy for further analysis.

Before describing how REDRAW analyzes structures, we explain the reasons for choosing this problem. An advantage of this domain for studying the role of visual reasoning in problem solving is the fact that it is rich with domain-specific knowledge that has significant implications on how the diagram is manipulated and interpreted. A number of researchers have studied pictorial reasoning in geometry, where pictures are abstract diagrams and do not represent anything in the real world [Koedinger and Anderson, 1990; Lindsay, 1992; McDougal, 1995]. In geometry, the only property one reasons about is the geometric property. There are no other types of information, apart from that represented in the diagram, that one must take into account when manipulating and inspecting the diagram.

In contrast, pictures used for reasoning in engineering design are not simply abstract geometric shapes but represent things in the real world. Furthermore, how a picture is interpreted and manipulated depends substantially on what it is meant to represent. For example, a line in our domain represents a beam or a column

Changing the curvature of a line would change the interpretation of the diagram. In a circuit diagram, on the other hand, one could change the length or curvature of the line representing an electrical connection without affecting its interpretation. For the goal of gaining an insight into the role of visual reasoning in problem solving and its relation to symbolic reasoning, it is important for us to work with a problem requiring a wealth of domain knowledge that has significant influence on the way diagrams are used and interpreted.

3 System Architecture

From examining the way deflection shape problems are solved by humans, it is apparent that solving this type of problems requires not only an ability to manipulate and inspect diagrams but also substantial structural engineering knowledge. Knowledge about the properties of various types of joints and supports is necessary to identify constraints on the shape for the structure to be in equilibrium. While such knowledge is best represented and manipulated symbolically, information about shapes is best represented as a picture. Many types of modification and inspection of the shape are also more easily carried out with a picture.

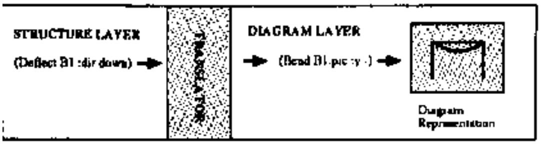
The requirement for both pictorial and non-pictorial representation and reasoning suggests a layered architecture. Thus, REDRAW includes both a symbolic reasoning component, called the Structure Layer, and a diagrammatic reasoning component, called the Diagram Layer. Figure 2 shows the two-layered architecture schematically. The Structure Layer contains the knowledge base of structural engineering knowledge about various types of structural members, joints, supports, and the constraints they impose on the structure. It also includes a rule-based inference mechanism to make use of the knowledge. The Diagram Layer includes an internal representation of the two-dimensional shape of the frame structure as well as a set of operators to manipulate and inspect the shape.

The Structure Layer contains a symbolic representation of domain-specific knowledge. It represents non-visual information, such as types of joints and equilibrium conditions, as well as heuristic knowledge for controlling the structural analysis process.

The Diagram Layer represents a diagram of the two-dimensional shape of a structure. It has operators that, directly act on this representation to allow inspection as well as transformation of the shape. 'These operators correspond to the manipulation and inspection operations people perform frequently and easily with diagrams while solving deflected-shape problems.

The internal representation of a diagram is a combination of a bitmap whose elements correspond to each "point" in a picture, and a more symbolic representation where each line is represented by a set of coordinate points.

The relations among the diagrammatic objects are quite straightforward. The objects relate to each other in qualitative spatial terms such as connected-to, left, right, above and below. Moreover, only those primitive geometric properties, such as whether the angle between two lines them is acute, obtuse or right angle, that are



Structure Layer
- **Objects**: beam, column, joint, support, load, etc.
- **Operators**: generate-constraint-for-hinge-joint, generate-constraint-for-fixed-support, etc.

Diagram Layer
- **Objects**: line, spline, point, etc.
- **Operators**:
  - **Manipulation**: rotate, bend, translate, smooth, etc.
  - **Inspection**: get-angular-displacement, get-displacement, etc.

Figure 2: Two-layered architecture of REDRAW

easily identified by visual inspection are explicitly represented.

There is a close link between the two layers. The system relates the representation of a particular beam in the Structure Layer to a spline in the Diagram Layer, and the concept of deflection of a beam to an operation on a spline to transform its shape. Likewise, the system iS able to identify features of a line (e.g. direction of bending, existence of an inflection point) and to communicate these features to the Structure Layer.

Communication between the two layers takes place by sending commands and posting constraints by the Structure Layer, which is carried out or checked by the Diagram Layer. There is a translator to mediate the communication. When the Structure Layer posts a constraint or a command, the Translator translates it into a call to a Diagram Layer operator that can directly act on the representation of the diagram to manipulate or inspect it. The result is again translated back to concepts that the Structure Layer understands.

REDRAW has successfully analyzed six of the 23 basic deflected shape problems described by Allen [Allen, 1978]. An informal evaluation by a civil engineer shows that the program reflects the qualitative reasoning process used in analyzing frame structures, and that it would be useful in helping students and novice engineers learn to solve problems of this type.

3.1 Example

This section describes REDRAW's problem solving process with the example in Figure 3, which is the same problem as the one shown in Figure 1.

Given the frame structure of Figure 3(a) and load, *Load3,* the Structure Layer, *SL,* sends a command, "Deflect *Beam3* in the same direction as the load", which the Translator, *Tr,* translates into an operation "Bend *Beam3.pic*[1] in the negative direction of the y-coordinate." Carrying out this operation will result in the shape shown in Figure 3(b). *SL* infers that since

[1] *X.pic* is a reference to the *DL* object that represents the shape of the *SL* object, *X*.

Joints is a rigid joint, *BeamZ* and *Column3* must remain perpendicular to each other at *Joint3.* *SL* issues a query to test this constraint. The query is translated into "Get the angle between *Beam3pic* and *Column3.pic,* at the ends connected by *Joint3.pic* for the Diagram layer, *DL.* The answer, the actual angle between the two lines, is communicated to *SL* as the answer that the constraint is not satisfied. *SL* now issues a command to satisfy this constraint while keeping *Beam3* fixed, which is translated into "Make the angle between *Beam3.pic* and *Column3.pic* at *Joint3.pic* 90 degrees without modifying *Beam3.pic"* for *DL.* Carrying out the operation will result in the shape shown in Figure 3(c). Communication will continue in this manner until all the constraints are satisfied.

3.2 Discussion

The design of REDRAW is greatly influenced by the ideas of Kosslyn [Kosslyn, 1980], and Chandrasekaran & Narayanan [Chandrasekaran and Narayanan, 1992] regarding human cognitive architectures, in which they argue that some types of reasoning are tightly coupled with perception. This idea of "perceptually grounded reasoning" is reflected in the architecture of REDRAW, which consists of symbolic and diagrammatic layers that are closely coupled.

The problem solving approach of REDRAW is designed to mimic the qualitative structural analysis method of human engineers. From the user's point of view, the design of the program allows one to concentrate on qualitative features of the structure, without requiring the specification of details. The diagrammatic components of the system facilitate the visualization of the particular deformation problem and its likely range of solutions. To aid in this visualization, the Diagram Layer operators include a "write-over" ability; that is, after a shape transformation, dotted lines show the original structure, just as a person draws a deformation right over the original line rather than create a separate new drawing. Displaying the shapes before and after each deformation step allows him to visually inspect and verify the inference process that was used in the shape transformation.

Since REDRAW follows a very informal analysis method, it does not work well when the structure becomes more complicated. Its most critical shortcoming is that it does not reason explicitly about forces and bending moments underlying the heuristics for determining how the deformation should propagate. As a result, when there are ambiguities due to multiple constraints being applicable to propagate deformation, it cannot resolve the ambiguity or even reason about the real cause of such ambiguity. Furthermore, because there is no explicit reasoning about forces and moments, one cannot proceed directly from the informal analysis to a more formal analysis, which would require reasoning about those concepts. To address these shortcomings, we undertook a second implementation of REDRAW, which is discussed in the following section.

4 REDRAW-II

REDRAW-II solves the same type of structural analysis problems as REDRAW, employing a more formal analysis technique involving explicit reasoning about force equilibrium, moment equilibrium and the shape, which the structural engineer must consider simultaneously in order to determine the final deflected shape. REDRAW-II draws and manipulates three different diagrams, a force diagram, a bending moment diagram, and a deflected shape diagram, to reason about the subproblems and display the results.

Because of the more formal analysis method REDRAW-II employs, REDRAW-II's problem solving capability is more general and also more easily extensible than that of the first implementation. However, with respect to diagrammatic reasoning, REDRAW-II relies more on symbolic reasoning to solve problems ihan REDRAW did. In REDRAW-II, the diagrammatic information is used more for control purposes than for actually solving a problem, especially for computing the forces and the bending moments. For example, forces are computed by rules in a few of the simplest cases, and by setting up and solving equations for the rest. For a given force, there are a large number of equations that can potentially produce an answer since one can formulate force and moment equilibrium equations for any subpart of the structure containing the location of the force. REDRAW-II has heuristics that use diagrammatic information, such as the position of a load on tinstructure or the deformation of a particular member, to determine what equation is most likely to produce an answer for a particular force given the available information.

4.1 Domain-Dependence of the Diagram Layer

In implementing REDRAW and REDRAW-II, we initially intended all the diagrammatic operators, such as bend, rotate and smooth, to be domain- and task-independent. However, it has become clear that while some operators are domain-independent, others are quite domain- and task-specific. For example, the implementation of the "bend" operator reflects the assumptions implicit in the domain and the task that the curvature of the bent line is large enough so that it can be clearly seen, but not so large that the structural member would appear to be broken.

Also, the inspection operators we have chosen to implement reflect the nature of the problem and of the domain knowledge we used. For example, most of the constraints in the structure layer are local constraints that apply to a single structural member or a set of members that are connected by a single joint. Thus, the inspection operators we have implemented are only those that detect local features of the diagram. In structural analysis. some global features such as symmetry are very useful, though we did not implement the operators for them. A more general-purpose diagrammatic reasoning layer will require a larger set of operators, which must be parameterized to work in a variety of situations. The set must include operators for inspecting and manipulating both local and global features of a diagram and must cover

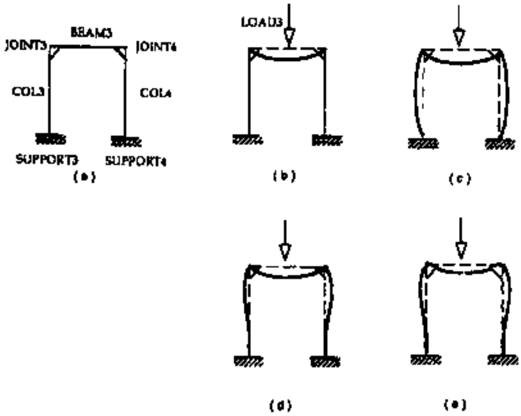


Figure 3: REDRAW solution to frame structure problem sketched in Figure 1

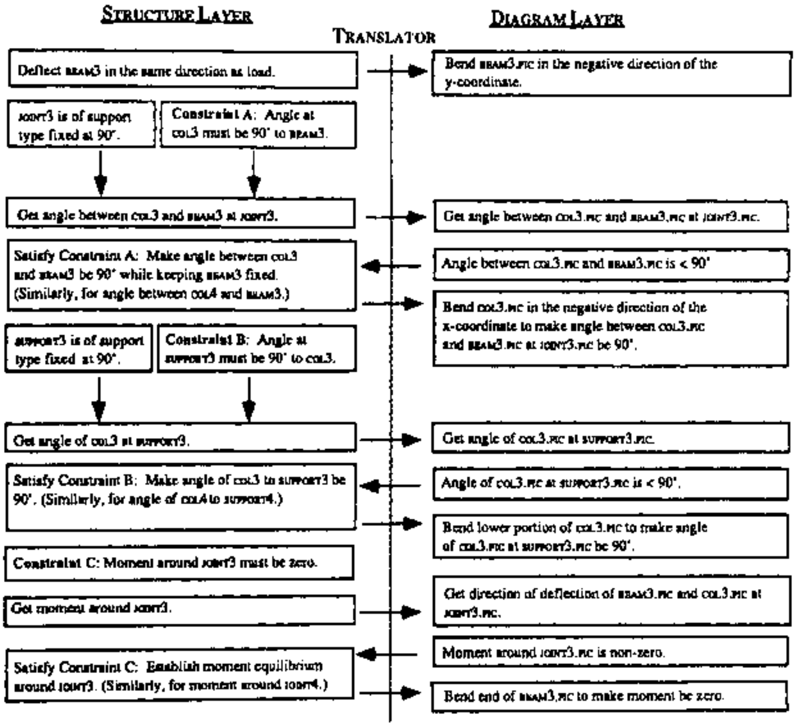


Figure 4: Illustration of the inter-layer communication of REDRAW for the example problem shown in Figure 3.

the types of operations humans perform frequently and easily with diagrams of many different types.

5 Related Work

We have previously built a program called QStruc to solve the same deflected shape problem described in this paper, but using a traditional, symbolic AI approach [Fruchter *et al.*, 1991]. There is no explicit representation of the shape of a structure in the program. The shape is implicitly represented by the existence of such physical processes as bending, and the qualitative values (positive, negative, zero or unknown) of such parameters as displacements. Both REDRAW and REDRAW-II solve problems more efficiently than QStruc. Their efficiency is due to the fact that diagrams allow the systems to focus the solution process much better than QStruc, which blindly sets up all applicable equilibrium equations and tries to solve them. Our informal evaluation of the systems also shows that REDRAW programs are much more instructive in helping the user to gain intuitive understanding of the physical phenomenon.

One of the first pieces of work that took the diagram as an integral part of understanding and solving a physical problem is Novak's work on physics problem solving. His system, ISAAC, solved problems in elementary dynamics[Novak, 1977]. ISAAC read a problem stated in English, generated an internal geometric model of the situation, set up mathematical equations, and solved them to produce an answer. It also drew a diagram to represent the given situation based on the geometric model. Though ISAAC did not actually use the diagram for problem solving, it used it to demonstrate its understanding of the problem to the user.

In Chandrasekaran and Narayanan's work on commonsense visual reasoning [Chandrasekaran and Narayanan, 1990], they proposed a visual modality-specific architecture, using a visual representation scheme, consisting of symbolic representations of the purely visual aspects (shape, color, size, spatial relations) of a given situation at multiple levels of resolution. The visual representation is linked to an underlying analogical representation of a picture so that visual operations performed on the analogical representation are immediately reflected on the visual representation and vice versa. Chandrasekaran and Narayanan's objective was "to propose a cognitive architecture underlying visual perception and mental imagery that explains analog mental imagery as well as symbolic visual representations" [Chandrasekaran and Narayanan, 1990].

Among the researchers of qualitative physics, Forbus was the first one to note the importance of diagrams in solving spatial problems. In his work on qualitative kinematics, he proposed the MD/PV model for representing spatial information [Forbus, 1980; Forbus *et al.*, 1991]. His MD/PV model consists of the Metric Diagram, which contains enough quantitative information to compute geometric features necessary for reasoning, as well as a Place Vocabulary, which is a set of relations that are appropriate for qualitatively representing (and solving) the particular problem at hand. Though, on the surface, our two-layered architecture with the Diagram and Structure Layers seems very similar to the MD/PV model, there are important differences. On one hand, Forbus' Metric Diagram is intended to represent quantitative information of a physical (spatial) situation, and the Place Vocabulary is a qualitative abstraction of the information in the Metric Diagram. On the other hand, our Diagram Layer is intended to represent diagrams used by people for solving problems. Whatever physical interpretation that the diagram may have is bestowed upon it by the Translator, which relates diagrams to the concepts in the problem domain. The information in the Structure Layer is not an abstraction of the diagram, but is the conceptual knowledge of the domain of structural analysis required to solve the deflection problems. This difference probably reflects the difference in focus: the main focus of our work is on the use of diagrams as essential medium of problem solving while Forbus' focus seems to be on solving qualitative kinematics problems. Whether this difference will result in diiferent, degrees of re-usability of the architectures (e.g. relative ease of reusing the problem solving architectures, especially the Metric Diagram and the Diagram Layer, for other domains) remains to be seen.

The work by DeCuyper and his colleagues[Decuyper *cl al.*, 1995], as well as an earlier piece of work by Gardin and Meltzer [Gardin and Meltzcr, 1989], both take a very different approach to reasoning about liquid from those based on symbolic qualitative reasoning. Instead of representing a body of liquid or a solid object as one entity as is usually done in symbolic reasoning systems, they represent both types of things as a one- or two-dimensional collection of particles. Each particle represents a small piece of liquid or solid stuff. They use a two-dimensional array to represent the position of each piece, and simulate the movement of each piece to predict the behavior of the collection. For simulation of the movement, DeCuyper et al. apply physics laws to each cell, while Gardin and Meltzer use local rules, which gov* ern the exchange of messages between neighboring particles. By changing the rules restricting the permissible angle between particles, Gardin and Meltzer can also simulate the behavior of solid objects, such as rods and rings, of different flexibility. As with Forbus' Metric Diagram, their analogical representation is not intended u> be a representation of a diagram but a representation of a model, composed of particles, of a physical situation. Their approaches seem promising, especially for reasoning about highly deformable objects. For such problems, relatively simple diagrams such as those used by REDRAW may not be be very useful, and those that do reflect the situation fairly accurately may be difficult to draw or to manipulate.

6 Summary and Conclusion

We have described our research on understanding the role of visual reasoning in a concrete problem-solving context. We have built prototype programs that reason qualitatively using diagrams in the same way that people do. Our decision to work with the deflection of shape problem in the domain of civil engineering was based on two considerations: First, since we had already built a system to solve the deflection problem using a traditional symbolic approach, we could directly compare the diagrammatic and symbolic reasoning approaches; and secondly, this was a knowledge-rich, real-world domain, which would allow us to study the role of diagrammatic reasoning in solving problems that required both types of reasoning.

In addition to examining the role of diagrammatic reasoning in problem solving, we are considering the generality of our work and its extendibility to other areas of technical design such as in architecture and mechanical engineering. Larkin and Simon [Larkin and Simon, 1987] showed that even with a symbolic representation, problem solving efficiency in some cases can be greatly improved by organizing the information in a way that reflects the physical structure of the object represented. By developing a strong understanding of the role that visual reasoning plays in the overall problem-solving process, we hope to construct a general tool that can be used to build diagrammatic reasoning systems in many other problem domains.

Acknowledgments

This research was supported by the Center for Integrated Facility Engineering of Stanford University as well as by the National Science Foundation, NSF Grant No. IRI-9408545.

References

[Allen, 1978] R. Allen. Elementary deflected structural theory. Department of Civil Engineering, Stanford University, 1978.

[Borning, 1979] Allan H. Borning. *Thinglab - a Constraint-Oriented Simulation Laboratory.* PhD thesis, Department of Computer Science, Stanford University, Stanford, CA, 1979.

[Brohn, 1984] David Brohn. *Understanding Structural Analysis.* Oxford, BSP Professional Books, Oxford, UK, 1984.

[Chandrasekaran and Narayanan, 1990] B. Chandrasekaran and Hari Narayanan. Towards a theory of cornrnonsense visual reasoning. In *Foundations of software, technology and theoretical computer science.* Springer-Verlag, 1990.

[Chandrasekaran and Narayanan, 1992] B. Chandrasekaran and Hari Narayanan. Perceptual representation and reasoning. In ,4,4,4/ *Symposium on Reasoning* ***with*** *Diagrammatic. Representations,* 1992.

[Chandrasekaran *et ai,* 1993] B. Chandrasekaran, Hari Narayanan, and Yumi Iwasaki. Reasoning with diagrammatic representations: A report on the AAA1 spring symposium. *AI Magazine,* 14:49-56, 1993.

[Decuyper *et ai,* 1995] J. Decuyper, D. Keymeulen, and L. Steels. A hybrid architecture based on a fine grained simulation as a model of liquids. In Janice Glasgow, Hari Narayanan, and B. Chandrasekaran, editors, ***Diagrammatic Reasoning:*** *Computational and* ***Cognitive Perspectives*** *on* ***Problem*** *Solving with Diagrams.* AAAI Press, Menlo Park, CA, 1995.

[Forbus *et ai,* 1991] Kenneth D. Forbus, Paul Nielsen, and Boi Faltings. Qualitative spatial reasoning: The clock project. *Artificial Intelligence,* 51(1-3):417-471, 1991.

[Forbus, 1980] Kenneth D. Forbus. Spatial and qualitative aspects of reasoning about motion. In *Proceedings of the First National Conference on Artificial Intelligence,* 1980.

[Fruchter *et al,* 1991] Renate Fruchter, Kincho H. Law, and Yumi Iwasaki. Qstruc: an approach for qualitative structural analysis. In *the Second International Conference on the Application of Artificial Intelligence to Civil and Structural Engineering.* Civil-Comp Press, 1991.

[Gardin and Meltzer, 1989] Francesco Gardin and Bernard Meltzer. Analogical representation of naive physics. *Artificial Intelligence,* 38(2):139-159, 1989.

[Koedinger and Anderson, *1990]* Kenneth Koedinger and John R. Anderson. Abstract planning -md perceptual chunks: Elements of expertise in geometry. *Cognitive Science,* 14:511-550, 1990.

[Koendinger, 1992] K. Koendinger. Diagrammatic reasoning by simulation. In *AAA J Symposium on Reasoning with Diagrammatic Representations,* 1992.

[Kosslyn, 1980] St .even M. Kosslyn. *Image and Mind.* Harvard University Press, Cambridge, MA, 1980.

[Larkin and Simon, 1987] Jill H. Larkin and Herbert A. Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science,* 11, 1987.

[Lindsay, 1992] Robert K. Lindsay. Diagrammatic reasoning by simulation. In *AAAI Symposium on Reasoning with Diagrammatic Representations,* 1992.

[McDougal, 1995] Thomas F. McDougal. Using diagrammatic features to index plans for geometry theorem-proving. In Janice Glasgow, Hari Narayanan, and B. Chandrasekaran, editors, *Diagrammatic Reasoning: Computational and Cognitive Perspectives on Problem Solving with Diagrams.* AAAI Press, Menlo Park, CA, 1995.

[Novak and Bulko, 1992] Gordon S. Novak and William C. Bulko. Uses of diagrams in solving physics problems. In *AAAI Symposium on Reasoning with Diagrammatic Representations,* 1992.

[Novak, 1977] Gordon S. Novak. Representation of knowledge in a program for solving physics problems. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence,* pages 286-291, 1977.